AMENDMENTS TO THE CLAIMS

1-48. (Cancelled)

49. (Currently Amended) A method of managing packet transmission in a computer system comprising:

receiving a plurality of packets from a computer host memory, wherein each packet has a header provided by a process running on a host processor;

reading at least one quality of service parameter from the header of each received packet;

storing each received packet into one of a plurality of queues according to the quality of service parameter, wherein each queue has a respective priority, wherein one of the plurality of queues is a high priority queue, wherein each queue of the plurality of queues other than the high priority queue has a corresponding timeout interval, and whereupon expiration of a timeout interval will cause a packet stored in the queue corresponding to the expired timeout interval to be forwarded ahead of packets stored in any other queue

dynamically allocating memory resources to each queue of the plurality of queues, comprising the steps of:

maintaining a list of free buffers for each queue, wherein each list of free buffers comprises a plurality of pointers to memory locations that are available to store packets assigned to the corresponding queue, wherein each pointer has a corresponding size parameter that specifies the size of the memory location indicated by the pointer; and

maintaining a list of used buffers for each queue, wherein each list of used buffers comprises a plurality of pointers to memory locations that are being used to store packets that have been assigned to the corresponding queue, wherein each pointer has a corresponding size parameter that specifies the size of the memory location indicated by the pointer, wherein each size parameter is programmable to allow for different buffers corresponding to the pointers in the plurality of free buffer lists and used buffer lists to vary in size; and

forwarding each received packet to a network medium according to both the priority of the queue in which the packet was stored and any expired timeout interval.

50. (Previously Presented) The method of claim 49 wherein receiving a plurality of packets from a computer host memory is performed by a Direct Memory Access download engine.

51. (Previously Presented) The method of claim 49 wherein storing each received packet into one of a plurality of queues comprises:
storing high priority packets into the high priority queue; and
storing low priority packets into a low priority queue.

52. (Previously Presented) The method of claim 49 wherein storing each received packet into one of a plurality of queues comprises:
storing high priority packets into the high priority queue;
storing intermediate priority packets into an intermediate priority queue; and
storing low priority packets into a low priority queue.

53. (Previously Presented) The method of claim 49 wherein forwarding each received packet according to both the priority of the queue in which the packet was stored and any expired timeout interval comprises:
preempting packet forwarding from the high priority queue and forwarding a packet stored in a lower priority queue when the timeout interval corresponding to the lower priority queue has expired.

54. (Previously Presented) The method of claim 49 wherein forwarding each received packet according to both the priority of the queue in which the packet was stored and any expired timeout interval comprises:
preempting packet forwarding from one or more higher priority queues and forwarding a packet stored in a lower priority queue when the timeout interval corresponding to the lower priority queue has expired.

55. (Previously Presented) The method of claim 49 wherein forwarding each received packet according to both the priority of the queue in which the packet was stored and any expired timeout interval comprises:

forwarding a packet stored in the high priority queue when the high priority queue contains packets to be forwarded and no time interval has expired;

preempting the high priority queue and forwarding a packet stored in an intermediate priority queue when the intermediate priority queue contains packets to be forwarded and the timeout interval corresponding to the intermediate priority queue has expired; and

preempting both the high priority queue and the intermediate priority queue, and forwarding a packet stored in a low priority queue when the low priority queue contains packets to be forwarded and the timeout interval corresponding to the low priority queue has expired.

56. (Previously Presented) The method of claim 49 wherein a received packet is compliant with the Ethernet protocol standard.

57. (Previously Presented) The method of claim 49 wherein a received packet is compliant with the Infiniband protocol standard.

58. (Previously Presented) The method of claim 49 wherein the plurality of queues correspond to a plurality of discrete storage arrays, wherein each discrete storage array of the plurality of discrete storage arrays corresponds to one of the plurality of queues.

59. (Previously Presented) The method of claim 49 wherein the plurality of queues correspond to a plurality of logical storage arrays and wherein each logical storage array of the plurality of logical storage arrays corresponds to one of the plurality of queues.

60. (Cancelled)

61. (Cancelled)

62. (Cancelled)

63. (Previously Presented) The method of claim 61 further comprising:

after forwarding a packet out of a queue, assigning the memory space associated with the forwarded packet from the used buffer list of the corresponding queue to the free buffer list of the one queue of the plurality of queues having the least quantity of free buffers in its free buffer list.

64. (Previously Presented) The method of claim 61 further comprising:

after forwarding a packet out of a queue, assigning the memory space associated with the forwarded packet from the used buffer list of the corresponding queue to the free buffer list of the one queue of the plurality of queues having the greatest quantity of packets forwarded over a measured timeframe.

65. (Currently Amended) A system for managing packet transmission in a computer system comprising:

a download engine for

receiving a plurality of packets from a computer host memory, wherein each packet has a header provided by a process running on a host processor;

reading at least one quality of service parameter from the header of each received packet; and

storing each packet into one of a plurality of queues according to the quality of service parameter;

a transmit packet buffer for maintaining the plurality of queues, wherein each queue has a respective priority, wherein one of the plurality of queues is a high priority queue, wherein each queue of the plurality of queues other than the high priority queue has a corresponding timeout interval, and whereupon expiration of a timeout interval will cause a packet stored in the queue corresponding to the expired timeout interval to be forwarded out of the transmit packet buffer ahead of packets stored in any other queue;

a plurality of free buffer registers, wherein each free buffer register corresponds to one of the plurality of queues, wherein each free buffer register maintains a list of free

buffers for its corresponding queue, wherein each list of free buffers comprises a plurality of pointers to memory locations in the transmit packet buffer that are available to store packets assigned to the corresponding queue, wherein each pointer has a corresponding size parameter that specifies the size of the memory location indicated by the pointer; and

a plurality of used buffer registers, wherein each used buffer register corresponds to one of the plurality of queues, wherein each used buffer register maintains a list of used buffers for its corresponding queue, wherein each list of used buffers comprises a plurality of pointers to memory locations in the transmit packet buffer that are being used to store packets that have been assigned to the corresponding queue, wherein each pointer has a corresponding size parameter that specifies the size of the memory location indicated by the pointer, wherein each size parameter is programmable to allow for different buffers corresponding to the pointers in the plurality of free buffer lists and used buffer lists to vary in size; and

a transmit engine for forwarding each received packet from the transmit packet buffer to a network medium according to both the priority of the queue in which the packet was stored and any expired timeout interval corresponding to any queue.

66. (Previously Presented) The system of claim 65 wherein the download engine:
stores high priority packets into the high priority queue; and
stores low priority packets into a low priority queue.

67. (Previously Presented) The system of claim 65 wherein the download engine:
stores high priority packets into the high priority queue;
stores intermediate priority packets into an intermediate priority queue; and
stores low priority packets into a low priority queue.

68. (Previously Presented) The system of claim 65 wherein the download engine is a Direct Memory Access download engine.

69. (Previously Presented) The system of claim 65 wherein the transmit engine:

preempts packet forwarding from the high priority queue and forwards a packet stored in a lower priority queue when the timeout interval corresponding to the lower priority queue has expired.

70. (Previously Presented) The system of claim 65 wherein the transmit engine:

preempts packet forwarding from one or more higher priority queues and forwards a packet stored in a lower priority queue when the timeout interval corresponding to the lower priority queue has expired.

71. (Previously Presented) The system of claim 65 wherein the transmit engine:

forwards a packet stored in the high priority queue when the high priority queue contains packets to be forwarded and no time interval has expired;

preempts the high priority queue and forwards a packet stored in an intermediate priority queue when the intermediate priority queue contains packets to be forwarded and the timeout interval corresponding to the intermediate priority queue has expired; and

preempts both the high priority queue and the intermediate priority queue, and forwards a packet stored in a low priority queue when the low priority queue contains packets to be forwarded and the timeout interval corresponding to the low priority queue has expired.

72. (Previously Presented) The system of claim 65 wherein a received packet is compliant with the Ethernet protocol standard.

73. (Previously Presented) The system of claim 65 wherein a received packet is compliant with the Infiniband protocol standard.

74. (Previously Presented) The system of claim 65 wherein the transmit packet buffer comprises a plurality of discrete storage arrays and wherein each discrete storage array of the plurality of discrete storage arrays corresponds to one of the plurality of queues.

75. (Previously Presented) The system of claim 65 wherein the transmit packet buffer comprises a plurality of logical storage arrays and wherein each logical storage array of the plurality of logical storage arrays corresponds to one of the plurality of queues.

76. (Previously Presented) The system of claim 75 wherein memory space in the transmit packet buffer is dynamically allocated to each queue of the plurality of queues.

77. (Cancelled)

78. (Cancelled)

79. (Previously Presented) The system of claim 77 further comprising:

a manager for assigning the transmit packet buffer memory space corresponding to a previously forwarded packet from the used buffer list of the queue corresponding to the previously forwarded packet to the free buffer list of the one queue of the plurality of queues having the least quantity of free buffers in its free buffer list.

80. (Previously Presented) The system of claim 77 further comprising:

a manager for assigning the transmit packet buffer memory space corresponding to a previously forwarded packet from the used buffer list of the queue corresponding to the previously forwarded packet to the free buffer list of the one queue of the plurality of queues having the greatest quantity of packets forwarded over a measured timeframe.

81. (Currently Amended) A computer readable media with instructions to cause a microprocessor to perform the steps of:

receiving a plurality of packets from a computer host memory, wherein each packet has a header provided by a process running on a host processor;

read at least one quality of service parameter from the header of each received packet;

storing each received packet into one of a plurality of queues according to the quality of service parameter, wherein each queue has a respective priority, wherein one of the plurality of queues is a high priority queue, wherein each queue of the plurality of queues other than the high priority queue has a corresponding timeout

interval, and whereupon expiration of a timeout interval will cause a packet stored in the queue corresponding to the expired timeout interval to be forwarded ahead of packets stored in any other queue;

dynamically allocating memory resources to each queue of the plurality of queues by performing the steps of:

maintaining a list of free buffers for each queue, wherein each list of free buffers comprises a plurality of pointers to memory locations that are available to store packets assigned to the corresponding queue, wherein each pointer has a corresponding size parameter that specifies the size of the memory location indicated by the pointer; and

maintaining a list of used buffers for each queue, wherein each list of used buffers comprises a plurality of pointers to memory locations that are being used to store packets that have been assigned to the corresponding queue, wherein each pointer has a corresponding size parameter that specifies the size of the memory location indicated by the pointer, wherein each size parameter is programmable to allow for different buffers corresponding to the pointers in the plurality of free buffer lists and used buffer lists to vary in size; and

forwarding each received packet to a network medium according to both the priority of the queue in which the packet was stored and any expired timeout interval.